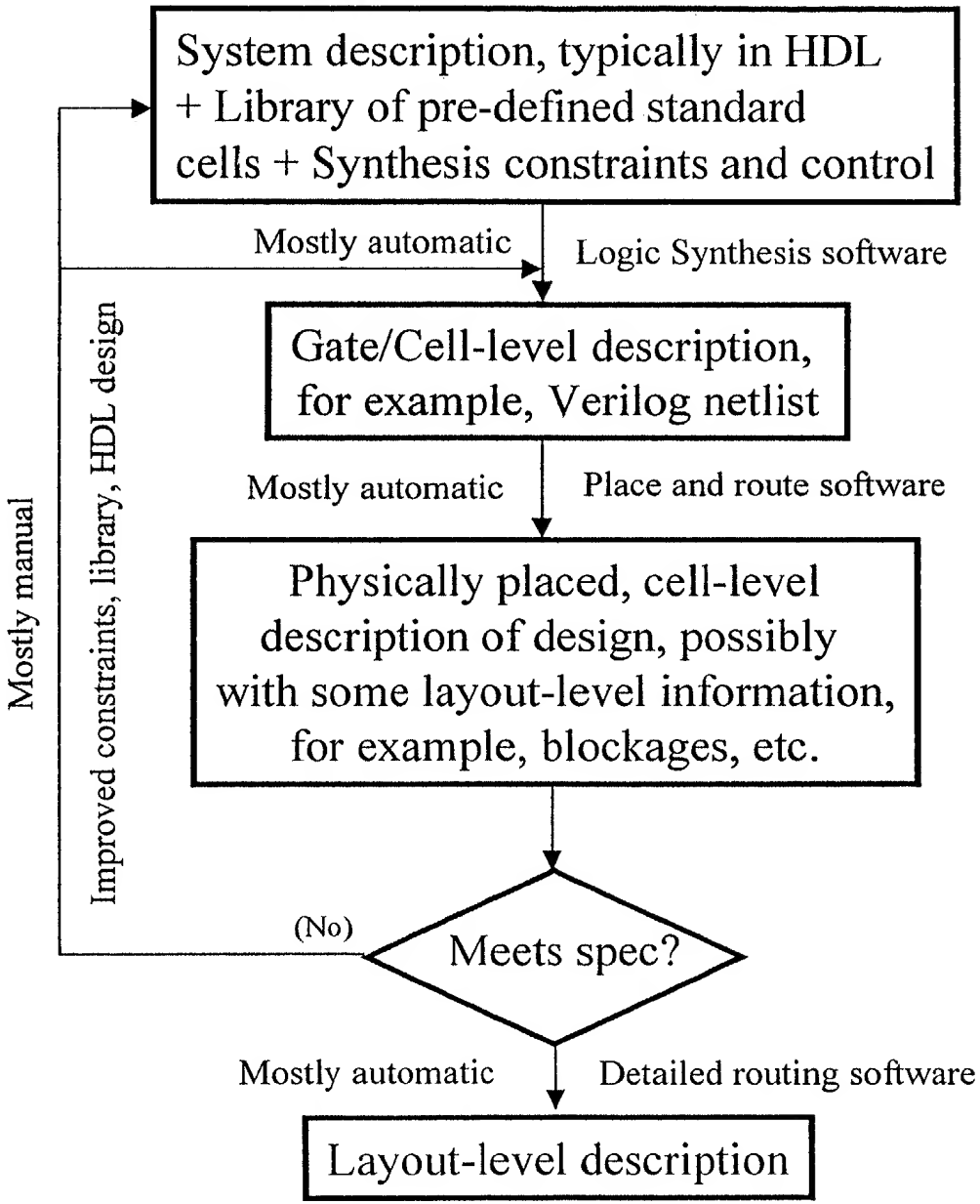

System description, typically in HDL + Library of pre-defined standard cells + Synthesis constraints and control
Mostly automatic
Logic Synthesis software
Gate/Cell-level description, for example, Verilog netlist
Mostly automatic
Place and route software
Physically placed, cell-level description of design, possibly with some layout-level information, for example, blockages, etc.
Meets spec?
(No)
Mostly manual
Improved constraints, library, HDL design
Mostly automatic
Detailed routing software
Layout-level description

Fig. 1

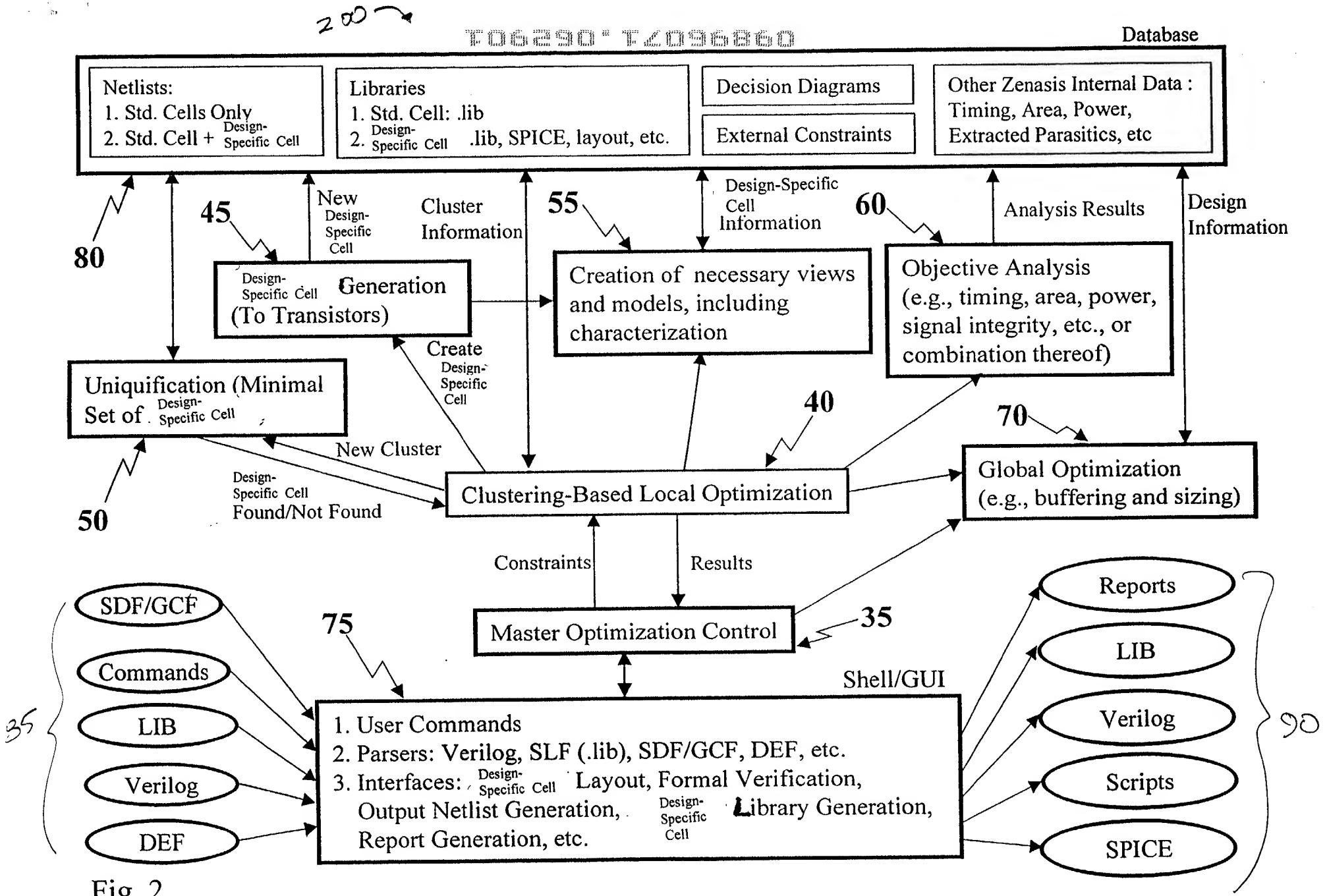

200
Database
Netlists:
1. Std. Cells Only
2. Std. Cell + Design-Specific Cell
Libraries
1. Std. Cell: .lib
2. Design-Specific Cell .lib, SPICE, layout, etc.
Decision Diagrams
External Constraints
Other Zenasis Internal Data : Timing, Area, Power, Extracted Parasitics, etc
80
45
New Design-Specific Cell
Cluster Information
55
Design-Specific Cell Information
60
Analysis Results
Design Information
Design-Specific Cell Generation (To Transistors)
Creation of necessary views and models, including characterization
Objective Analysis (e.g., timing, area, power, signal integrity, etc., or combination thereof)
Create Design-Specific Cell
Uniquification (Minimal Set of Design-Specific Cell
40
70
New Cluster
50
Design-Specific Cell Found/Not Found
Clustering-Based Local Optimization
Global Optimization (e.g., buffering and sizing)
Constraints
Results
SDF/GCF
Commands
LIB
Verilog
DEF
75
Master Optimization Control
35
Shell/GUI
1. User Commands
2. Parsers: Verilog, SLF (.lib), SDF/GCF, DEF, etc.
3. Interfaces: Design-Specific Cell Layout, Formal Verification, Output Netlist Generation, Design-Specific Cell Library Generation, Report Generation, etc.
Reports
LIB
Verilog
Scripts
SPICE
35
90

Fig. 2

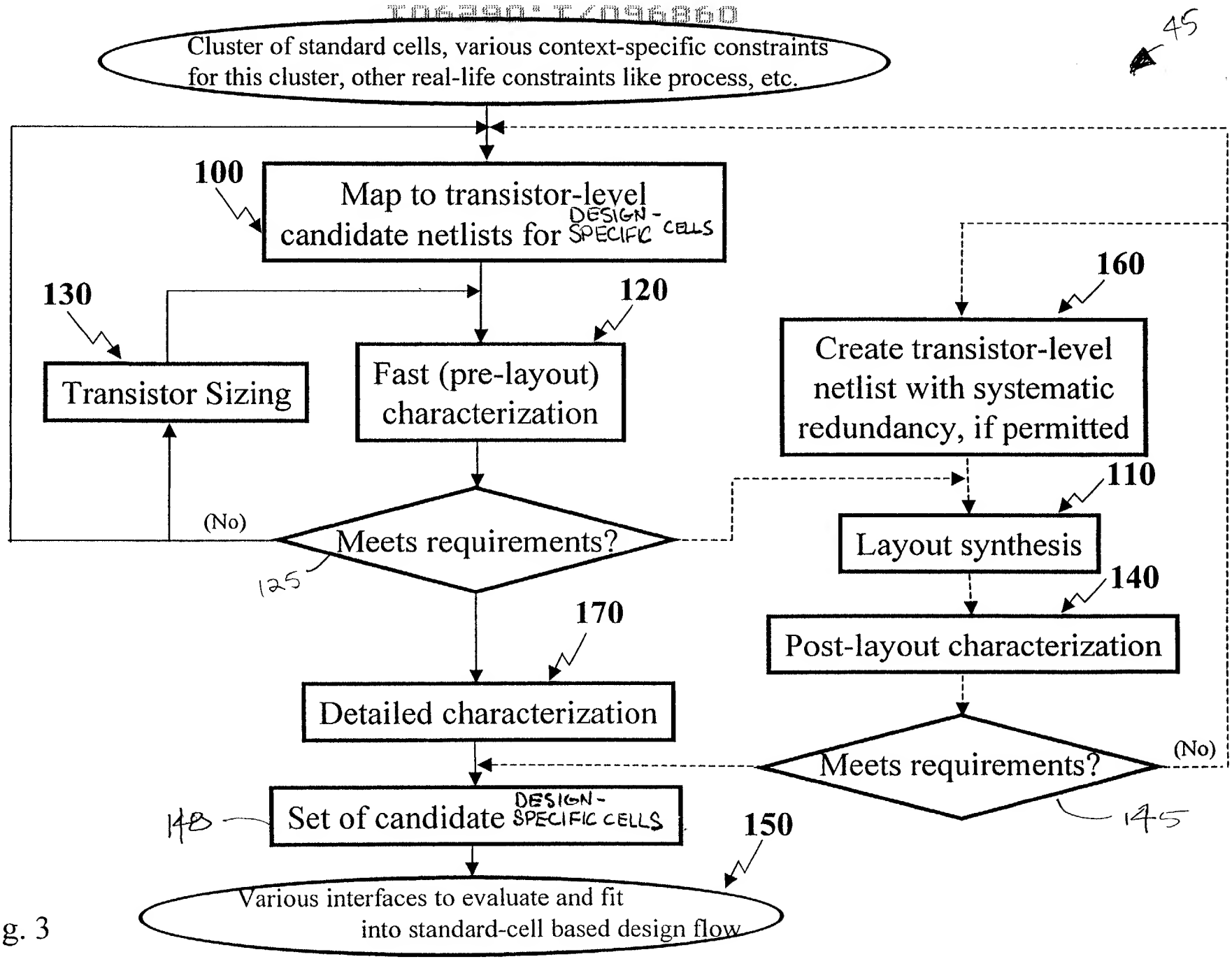

Cluster of standard cells, various context-specific constraints for this cluster, other real-life constraints like process, etc.
45
100
Map to transistor-level candidate netlists for DESIGN-SPECIFIC CELLS
130
Transistor Sizing
120
Fast (pre-layout) characterization
160
Create transistor-level netlist with systematic redundancy, if permitted
110
Layout synthesis
140
Post-layout characterization
(No)
Meets requirements?
125
170
Detailed characterization
Meets requirements?
(No)
145
148
Set of candidate DESIGN-SPECIFIC CELLS
150
Various interfaces to evaluate and fit into standard-cell based design flow

Fig. 3

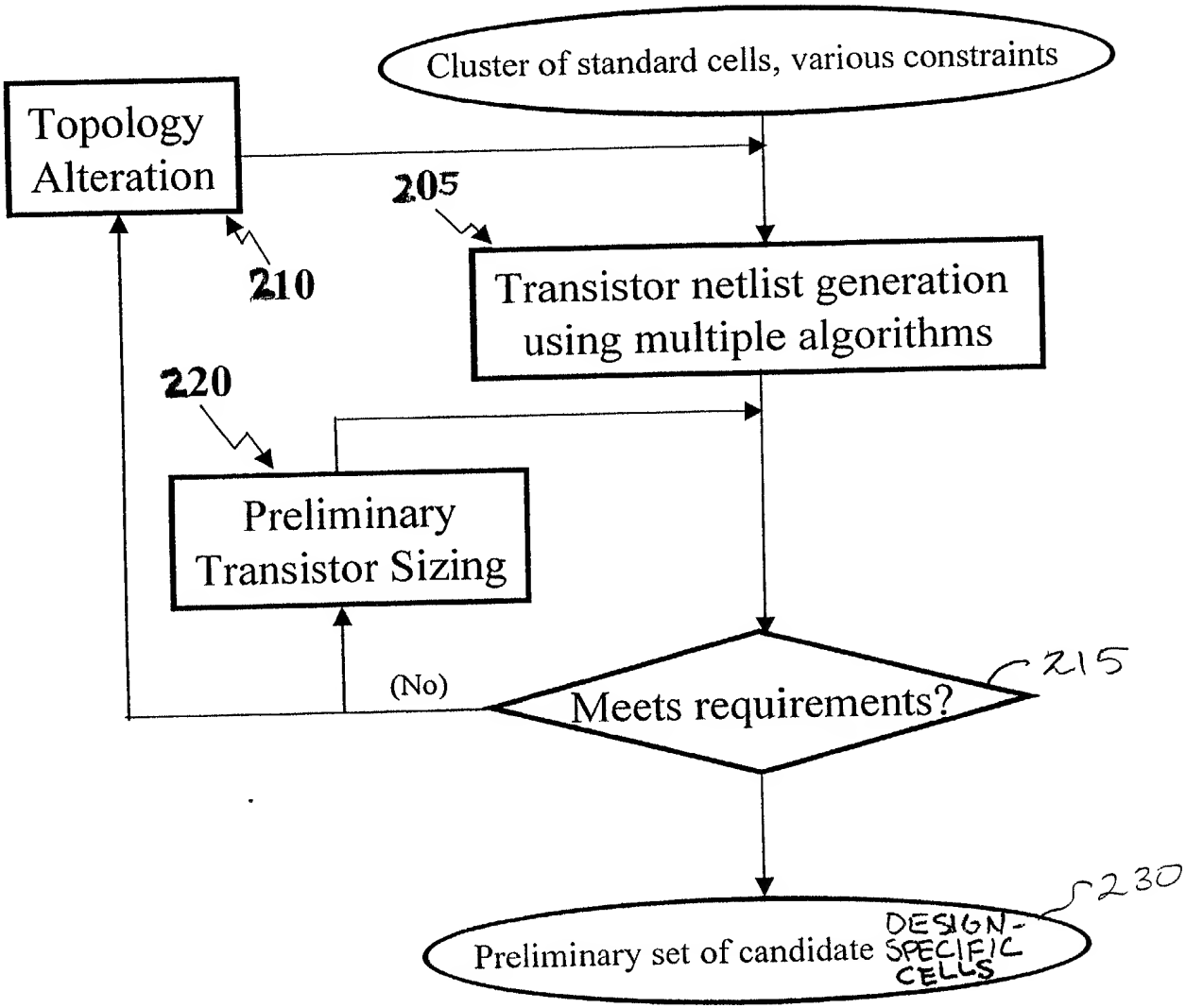
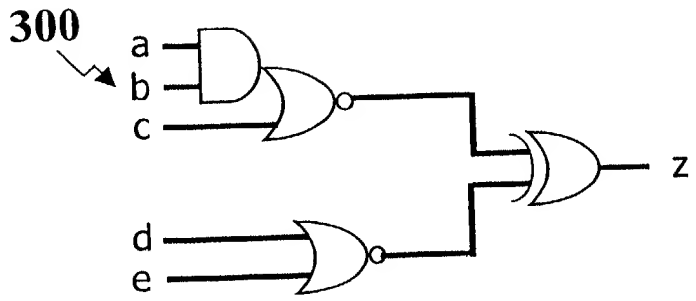


Fig. 4



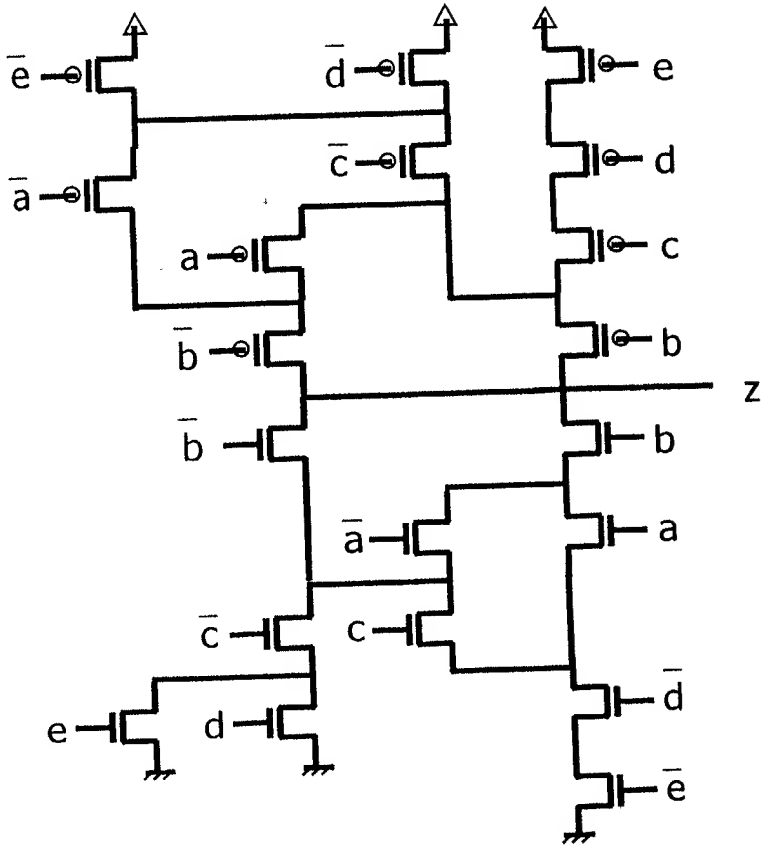
5a

Critical path to z through b

⇒ **Delay from b to z : 0.33 ns**

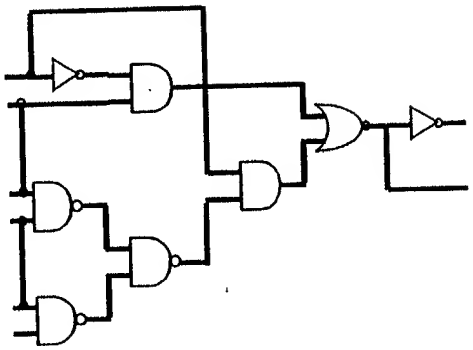
⇒ DESIGN-SPECIFIC CELL **delay : 0.14 ns**

5c

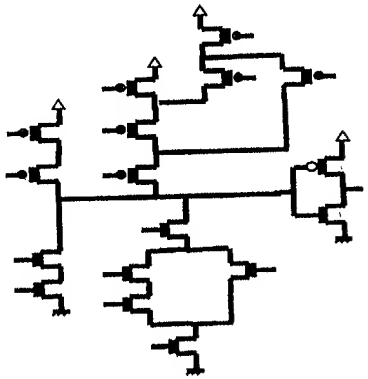


5b

Fig. 5



6a

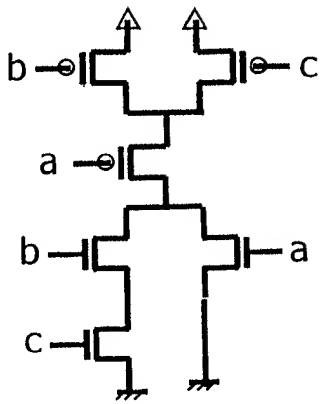


6b

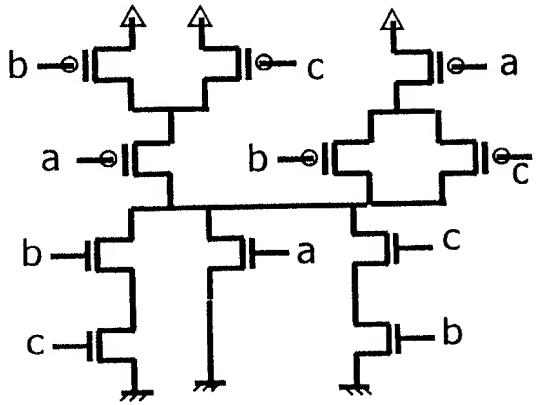
Criteria	Original	Optimized
#of Cells	8	1
#of Transistors	32	17
#of Wires (incl. I/O)	12	5

6c

Fig. 6



Typical standard cell implementation with no systematic redundancy -- input a usually has fastest propagation through cell, c slowest



Implementation of same functionality with systematic redundancy -- inputs a and c usually have comparable propagation delay through module

Fig. 7